Visual Search Modelling: A Deep Learning Approach on Graphical Layouts

Rishabh Kapoor
Dept. of Computer Science
Espoo, Finland
rishabh.kapoor@aalto.fi

ABSTRACT

Visual search modelling provides an excellent opportunity to predict the usability of a user interface design before even testing it. Furthermore, it provides a better understanding of user behaviour which can help UI researchers and designers to make better decisions while designing the UI. This paper presents a deep learning approach for visual search modelling which helps predict the scanability of a graphical layout. The idea behind using a deep learning approach is to make use of raw pixels for predicting scanability of a target element. The model provides a classification output where the targets have been labelled on the level of difficulty that a user might face in searching them in the UI.

Author Keywords

visual search, guis, hci, deep learning, visual modelling

INTRODUCTION

Visual search modelling is a topic of interest in the domain of Human-Computer Interaction [4, 39]. Modelling visual search will allow us to explore many avenues in User Interface Design and User Experience. A model that can predict the usability of a User Interface can save a huge amount of time and cost for UI Researchers and Designers. Furthermore, it will also provide insights on selecting the various aspects in the design like alignment, positioning, size etc which in general present a better design idea for the UI of their application. Traditionally UI testing is done by showing various samples of a design to a human tester and recording their reaction, this approach is not only time consuming but also offers little (and sometimes zero) insights on a design for a general audience. Hence, a model that can automate this process will not only save a large amount of time consumed in testing but will also present a generalized understanding of a design.

Many previous works have been done in the field of visual search that have formed a primitive basis for the understanding of visual search [38]. Visual Search can be classified into two categories: feature search and conjunction search, Treisman and Gelade gave the basis of feature search in their work, where they proposed that features like color, shape, size etc, help differentiate between targets and distractors. On the contrary conjunction search is described as separating a previously known target from distractors that share some visual features with the target [34, 24]. Studies have shown [31] that the number of distractors do not affect the efficiency i.e. reaction time and accuracy in the case of feature search while in the case of conjunction search if we increase the number of distractors then reaction time and accuracy will decrease. Studies [42] have shown that previous knowledge about the target greatly influences the visual search and the top-down search process is highly efficient rather than a bottom-up process. Treisman and Gelade [34] also introduced a Feature Integration Theory that discusses how some feature has a greater influence and are registered much earlier in the brain. Chan and Hayward [5] conducted different experiments demonstrating how visual searches performed in a single dimension are much more efficient and apparent while adding dimensions will lead to complexity.

The early methods perform well under laboratory or controlled conditions and thus they are not practical in realistic scenarios. An example of it can be understood from the works [6, 22] that have used straightforward features like geometrical shapes, the distractors are separable from the target element and hence these early findings provide little understanding of visual search modelling in real life.

Studies [18, 33] have modelled the visual search with the help of features like shape, size, color etc of the elements. This technique is very sound, however, there are a lot of inputs that are required to be fed to the model to learn. Moreover, this raises a problem where a new element or an interface is tried to be modelled. The difficulty of using these models on the unseen and the new dataset and providing inputs for new dataset has made it difficult to use these models on a wide range of User Interfaces.

However, the deep learning techniques have shown many promising results in diverse areas. Deep learning has been used in medical imaging [3, 37], image classification [2, 9, 10] and many other fields. An advantage of using a deep learning model is that it provides the learner the ability to capture the features and relations between different objects that may not be understood by humans easily. Hence, using deep learning based approaches could save a lot of time and resources used in feature engineering, this has made deep learning an excel-

CHI'20, April 25–30, 2020, Honolulu, HI, USA

ISBN 978-1-4503-6708-0/20/04…$15.00

DOI: https://doi.org/10.1145/3313831.XXXXXXX

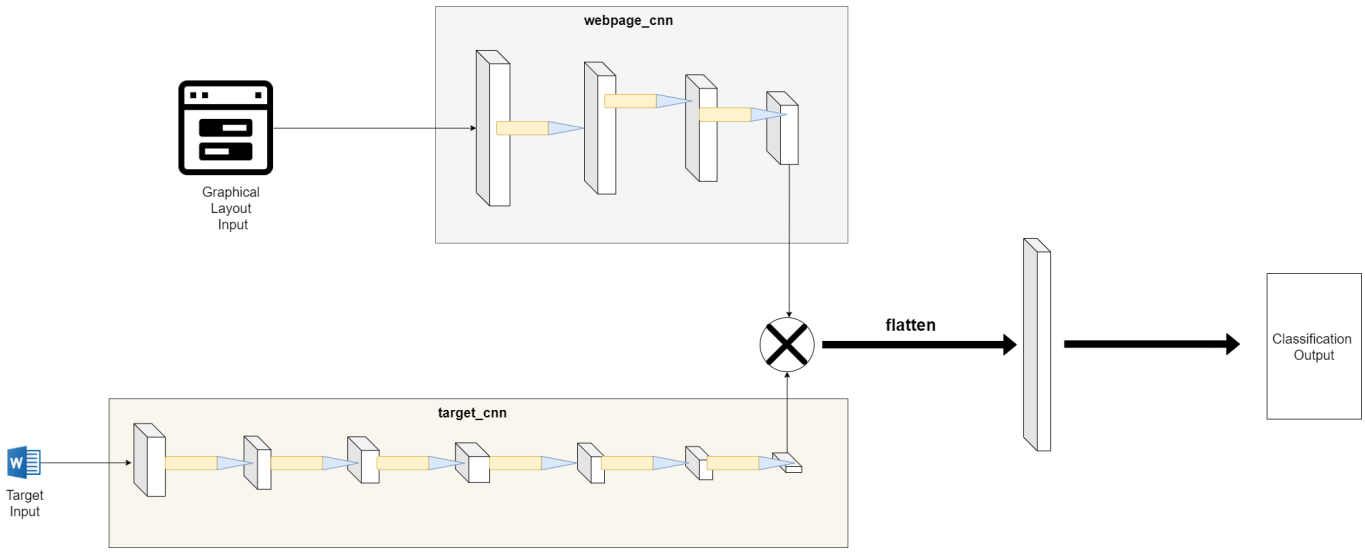

Figure 1. A schematic diagram of the architecture

lent tool in the domain of Human-Computer Interaction(HCI). Several latest researches have used deep learning [23, 29, 21] for solving various interaction problems as well as advancing performance modelling. This work presents the use of deep learning technique for modelling visual search. This work shows how can we input raw pixels from the image of the user interface to predict the visual search time of the element present in it.

The domain of user interfaces is being profoundly explored with deep learning. A lot of approaches have been proposed to fast track the development of design to the User Interface, [14, 30, 25] proposes using deep learning to transform sketches to the User Interface for quick mock-ups and to test implementations, but these are again based on design generations and hence may not be suitable for a wide audience.

The following are contributions of the paper:

- A deep learning model that can predict visual search time on realistic Graphical User Interfaces. This allows a lot of possibilities for modelling and encompass a wider set of conditions. A schematic diagram of the same has been given in Figure 1.
- The model inputs are raw pixels which not only provide a faster way for data ingestion but also save a lot of time in feature engineering for machine learning models. Figures 2 and 3 present an example of user interfaces used as input.
- An analysis and understanding of the behaviour of deep learning model that describes how the model behaves for a given set of inputs, how the attention map is computed and how it helps in the prediction.

The following sections of the paper are organized in the following manner:

- The Related Work section [Section 2] presents a survey of literature that have put a strong foothold in this area.
- The Methodology section [Section 3] presents the methodology followed in the course of research. The subsections present elaborated explanations on data collection, data augmentation and how the deep learning model was built and trained.
- The Results section [Section 4] presents results that were extracted during the experimentation and how can we analyse the result we got from the deep learning world
- The Conclusions and Discussions section [section 5] provides conclusions for our current model and techniques and what do we plan for any possible future research.

RELATED WORKS

Visual Search modelling is quite useful because it gives us the ability to predict the usability of the interfaces before their release. It also helps to develop an understanding of the human behaviour which could improve the user experience in all. However, modelling a visual search process has been a progressing investigation in the field of Human-Computer Interaction (HCI) [19, 8].

Itti and Koch models [16, 15] have proposed that attention is chiefly designated to the most salient visual zone i.e. the region that is distinguishable from its surrounding. The work [15] also incorporates the short-term memory component which prevents attention to the most salient regions.

Yuan and Li in their work [40] proposed a deep learning approach to predict human visual search time on large-scale realistic web pages. They claim that this approach can easily accommodate both structured and unstructured data which provides a good generalisation. Jokinen et al [18] presented a model on visual search modelling on graphical layouts. The model uses the fact that the perception is guided by long term memory. Hence, it assumes that the users gain more experience after repeatedly using the interface and this knowledge can be used to improve the user interface. Heinke and

Figure 2. Sample of User Interfaces used

Humphreys [12] proposed a model which they call SAIM that mimics a human ability to identify an object in multi-object scenario. However, they assume the object to be translation-invariant. The authors then extend this to use this model for visual search experiments [11]. They provide the results for some asymmetric and symmetric searches.

Elazary and Itti in their work [7] have proposed a bayesian technique for visual search. The statistical inference was used to perform visual search for the target object and give a probability of whether an object is a target or a distractor (an object that is not useful or may distract the user from finding the target). As this was a probabilistic method, this approach was quite simplistic but the results were good. The approach performed better and faster than its predecessor.

Nyamsuren and Taatgen[27] proposed a computational model which they termed as PAAV (Pre-attentive and Attentive Vision module) that uses bottom-up and top-down processing in visual search. The pre-attentive part derives the information about the stimuli from the visual automatically, the information includes the sizes, locations, colors and orientations of the target elements. The module also has a top-down attentive stage that is used to guide attention towards the elements that have similar visual features with the desired targets.

Jimenez et. al. in their work [17] has explored the use of Class Activation Maps (CAMs) that improves the fixed region sampling strategy of R-MAC. CAMs can generate spatial maps that highlight the contribution of the areas in an image that are crucial for classifying the image with a particular label. Lanskar and Kannala [20] made use of saliency measures to identify the contribution of the R-MAC regions to aggregation. Mohedano et. al. introduced a novel saliency prior [26] for aggregating local CNN features. They claim that the training can be done in an unsupervised fashion and sometimes it is not required at all.

Unlike the traditional approaches, the model proposed in this paper is not limited to specific visual tasks. This work aims to incorporate a large number of visual task concerning User Interfaces and not just one particular type of them. Moreover, the model is not dependent on any kind of structured input which has to be manually collected from an image which is extremely troublesome in some cases but uses raw pixels to model the visual search task.

The proposed model follows an approach similar to the attention mechanism [35], here attention is being focused on the target element which is to be modelled. Attention map helps to build a human-like intuition where the focus is made on the area which has the element. Analysing the attention map can also help us predict the areas where the primary focus lies hence it can offer more insight into the UI, but the scope of this work is limited to modelling the visual search time of elements in the UI.

METHODOLOGY

Dataset

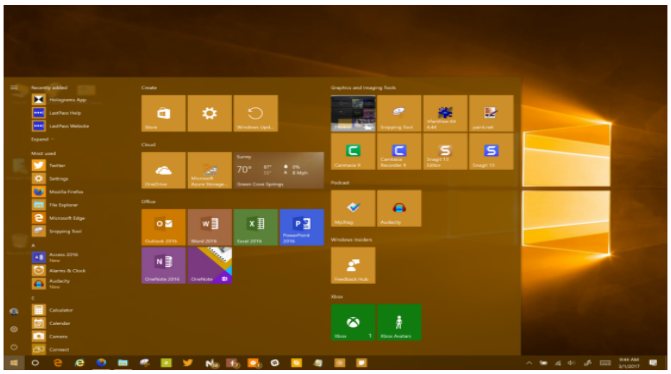

Figure 3. A sample of Windows 10 Desktop

The dataset to train and test the deep learning model is used from one used by Jokinen et. al. in their work[18]. The dataset contains various User Interface images in which different positioning of different elements are used as inputs. Figures 2 provides an example of different types of user interfaces that are fed to the model as input while Figure 3 showcases a user interface of the windows 10 operating system that has been used for input of the model.

Here, if we check Figure 3, we can see there are many variations of a single screen that will exist. Hence, to generalize the result the user interfaces have been shown to different people and their result has been annotated to achieve a common consensus of the elements that are deemed Hard or Easy.

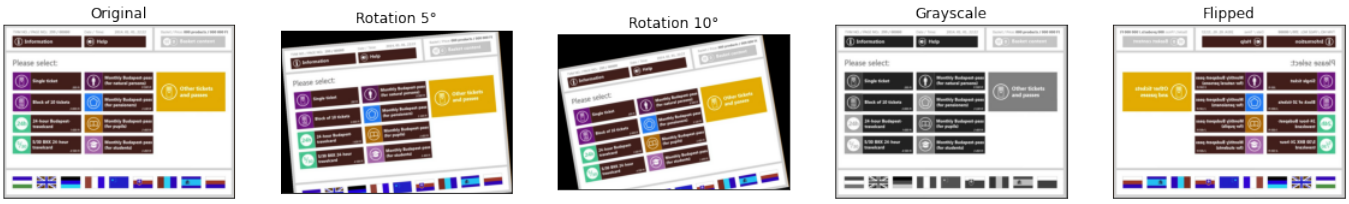

Figure 4. Data Augmentation Techniques Used

Upon exploration, the elements in a single UI may be divided into five categories namely *Very Easy*, *Easy*, *Neutral*, *Hard*, *Very Hard*.

The following is the meaning of the categorization:

- *Very Easy*, These are those elements that are very easy to identify in the user interface. The example includes the icons or text that are quite big in the UI or are very differentiable from others. They are instantly identified.
- *Easy*, These elements take a little time to be identified. These elements share characteristics with the previous category, however, these elements might not be common so as to be instantaneously identified.
- *Neutral*, In this category, the elements are not easily identified but not much stress is put on the user to identify them. They take more time than those categorized as easy but are still not stressful to detect.
- *Hard*, These elements are not very easily identified. At times the search had to be performed multiple times by the people to identify the elements. However, they are ultimately identified by the user.
- *Very Hard*, These elements are missed by a majority of users or either the users spent a huge amount of time or may require many different gazes to spot the elements.

However, a very important requirement of a deep learning model is to train the model on a huge set of data and to gain confidence a considerable amount of test set is required. Due to resource constraint, a Data Augmentation approach has been used to augment the data to expand the training set of the model.

In this paper, we define two elements, a source image which is the complete image of UI as shown in Figure 3 and a target image which is a snapshot of various elements of the same image.

Data Augmentation

Data augmentation has proven to be a very efficient way to incorporate more data for testing and training. Various researches have applied various data augmentation techniques to improve the dataset for training and to get more confidence in their model. Taylor and Nitschke [32] have shown various data augmentation techniques in their work. Along with the approaches they have shown the effectiveness of these methods on different algorithms. Wang and Perez in their work [28] have also performed a similar study to observe the effect of data augmentation on various algorithms.

The basic idea behind applying transformations to an image is that the structural integrity is maintained. The various relationships between different elements are not hindered and hence deep learning model can establish more correlations between images. To avoid overfitting of the model, the transformations are just 4 and they are applied to all the source input images as well as the target image.

Figure 4 shows the different types of transformations applied to the images in the dataset. The same transformation is also applied to the target elements to get the corresponding target. It has been made sure that there is no data loss in the input, so the target elements exist in the image i.e. the target elements that do not appear in the final image after transformation have been removed from the transformation data points.

Input Data Pipeline

The input of the deep learning model is simple, a raw image is provided as input along with a raw image of the target. However, storing images in memory are costly as they require a lot of storage in the memory. Also, we don't want to supply a variable resolution of the images as this might create an imbalance. Hence, it is important to design an Input data pipeline for efficient ingestion.

The ingestion starts with reading a source image from the directory and then reading the target images associated with them and finally reading the label. After the source image is read, it is re-scaled to 512×512 dimension to maintain consistency along with all the inputs. The target images are then resized to the dimension 64×64 to maintain consistency. The labels are encoded with one-hot encoding scheme represented in Table 1.

Label	Encoding
Very Easy	[1,0,0,0,0]
Easy	[0,1,0,0,0]
Neutral	[0,0,1,0,0]
Hard	[0,0,0,1,0]
Very Hard	[0,0,0,0,1]

Table 1. Encoding schemes for Labels.

With one hot encoding, we can train the model and the model will give predictions for different classes, this will not only ensure that the highest prediction might be the actual answer but can also help us approximate. For eg., if we receive output

for predictions saying the probability of Easy label is 0.4 and probability of Very Easy label is 0.42, we can use this result to model that the element is easy to search.

Model Hyperparameters

Figure 1 shows the representation of the model. The webpage_cnn consists of 3 convolutional layers each of kernel size 3×3 and output channels are 4. A batch normalization layer to stabilize and accelerate training along with a ReLU activation function and a max-pooling layer of size 2×2 exists between each convolutional layer. The final embedding computed by this network is of dimensions $64 \times 64 \times 4$.

The target_cnn has the same architecture but instead of 3 convolutional layers it has 6. The final embedding computed by this network is of dimensions $1 \times 1 \times 4$. After we receive embeddings from both networks we apply a cosine similarity between the target embedding and each super-pixel representation in the webpage embedding. The final saliency map is then 64×64 which is flattened to yield size of 4096 dimensions.

Deep Learning Model

The previous section has discussed the parameters used to build the model. The model has been implemented in the TensorFlow library [1]. To obtain the output the softmax activation function is applied on the output of the flatten layer to obtain the final result on a dense layer of 5 units. The output will be a probability score for each class. An adam optimizer with a learning rate 0.0001 and categorical crossentropy loss function [41] is used as the loss function. There are 10 epochs used for training.

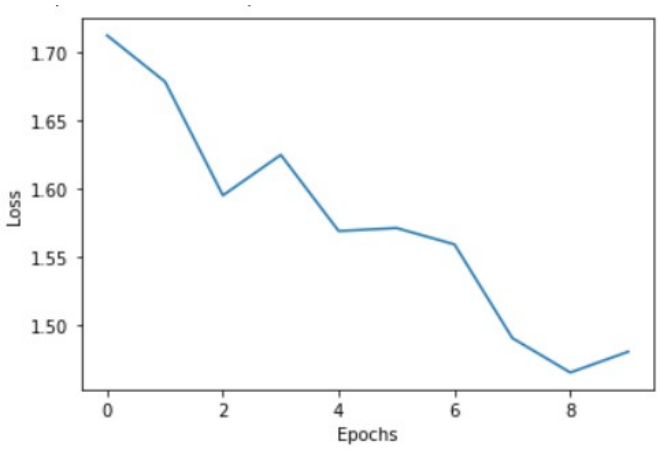

Figure 5. Training Loss in different Epochs

Figure 5 shows the efficiency of training i.e. how does the loss decrease with the epochs. The 10 epochs provide decent learning with no overfitting.

RESULTS

For the scope of this paper, a test set is defined, the test set includes similar source and target elements. However, the target elements are completely new to the model i.e. model has no prior information on them as they are not involved during training process. This gives us an opportunity to test the model with certain elements and to verify the result.

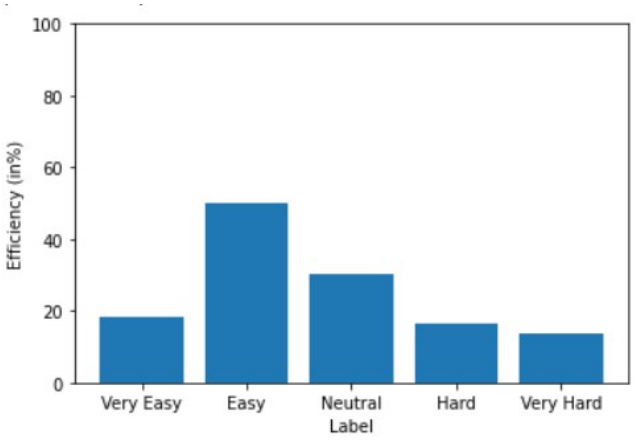

Figure 6. Accuracy for different classes

Overall, the model can correctly predict labels for 25.3% of cases, which is not very promising to implement this model in the real-life scenario. However, the train and test sets does not a contain good amount of images. Increasing images will actually provide better training and hence the model will be able to better predict the labels.

Figure 6 provides an overview of the performance of the model on the test data. The softmax activation function outputs to five nodes which correspond to the label we intend to predict. The model doesn't just gives us prediction for a label as such, but it gives a probability score for each label. So unlike the input label, the model doesn't gives 1 at the place but provides the probability score for each label. So the position with the highest score is taken to be as the label for the pair of test images. The actual accuracy from the model is reported to be 25.3% with the help of categorical crossentropy loss. However, interestingly for 46.1% of cases the output from the model is the adjacent label, this hence can be interpreted as the near approximation of the label, which signifies that for 71.5% of cases the labels are either correct or approximate. Figure 7 shows a bar plot for classes where the adjacent labels were predicted.

From the results (Figure 6 and Figure 7), it can be seen that the model performs well for all labels except for the label "Very Hard". One fact is that the quantity of data is not appropriate for the class. Moreover, the correct prediction of adjacent classes is also useful, because visual search will vary from person to person and hence even if we can approximate a class it can provide a rich insight for the UI.

CONCLUSIONS AND DISCUSSIONS

The paper has presented a deep learning approach to predict the visual search time on the graphical user interfaces. The convolutional neural network along with the attention mechanism has performed good, a lot of time that might have been spent on feature engineering or feature extraction has been saved and the new dataset could be easily incorporated into

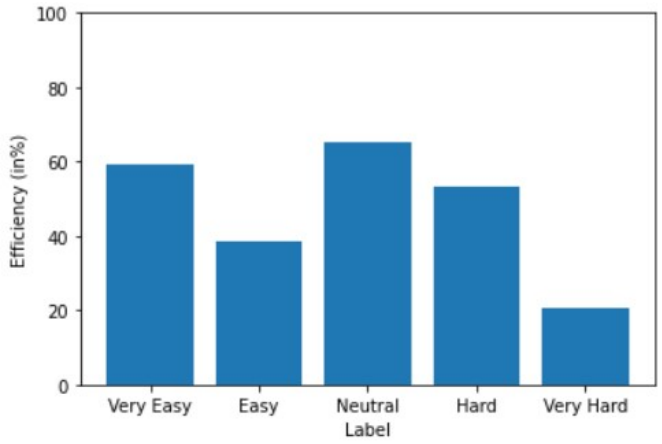

Figure 7. Accuracy for different classes when adjacent classes are predicted instead of correct class

this approach. However, the accuracy is far from good. To achieve better accuracy some of the following steps can be taken:

- Convolutional Neural Networks are known to have various limitations [13], datasets should have a large number of data points.
- Recurrent Neural Networks can be tried to check whether it provides better results than CNNs [36]. They can also incorporate the attention mechanism which is not possible in CNNs.
- The model has only made use of raw pixel data, to improve the results a large number of training samples should be included.
- The annotations are performed on a less number of people, hence the number of trials could be improved to generalize the viewpoint more.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). `https://www.tensorflow.org/` Software available from tensorflow.org.

[2] Görkem Algan and Ilkay Ulusoy. 2021. Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Systems* 215 (2021), 106771.

[3] Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Quoc-Viet Pham, Thippa Reddy Gadekallu, Chiranji Lal Chowdhary, Mamoun Alazab, Md Jalil Piran, and others. 2021. Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey. *Sustainable cities and society* 65 (2021), 102589.

[4] Ali Borji. 2019. Saliency prediction in the deep learning era: Successes and limitations. *IEEE transactions on pattern analysis and machine intelligence* (2019).

[5] Louis KH Chan and William G Hayward. 2013. Visual search. *Wiley Interdisciplinary Reviews: Cognitive Science* 4, 4 (2013), 415–429.

[6] J Edwin Dickinson, Krystle Haley, Vanessa K Bowden, and David R Badcock. 2018. Visual search reveals a critical component to shape. *Journal of vision* 18, 2 (2018), 2–2.

[7] Lior Elazary and Laurent Itti. 2010. A Bayesian model for efficient visual search and recognition. *Vision research* 50, 14 (2010), 1338–1352.

[8] Tim Halverson and Anthony J Hornof. 2011. A computational model of "active vision" for visual search in human–computer interaction. *Human–Computer Interaction* 26, 4 (2011), 285–314.

[9] Yanling Han, Yekun Liu, Zhonghua Hong, Yun Zhang, Shuhu Yang, and Jing Wang. 2021. Sea Ice Image Classification Based on Heterogeneous Data Fusion and Deep Learning. *Remote Sensing* 13, 4 (2021), 592.

[10] Mohd Anul Haq, Gazi Rahaman, Prashant Baral, and Abhijit Ghosh. 2021. Deep Learning Based Supervised Image Classification Using UAV Images for Forest Areas Classification. *Journal of the Indian Society of Remote Sensing* 49, 3 (2021), 601–606.

[11] Dietmar Heinke and Andreas Backhaus. 2011. Modelling visual search with the selective attention for identification model (vs-saim): a novel explanation for visual search asymmetries. *Cognitive computation* 3, 1 (2011), 185–205.

[12] Dietmar Heinke and Glyn W Humphreys. 2003. Attention, spatial representation, and visual neglect: simulating emergent attention and spatial memory in the selective attention for identification model (SAIM). *Psychological review* 110, 1 (2003), 29.

[13] Hossein Hosseini, Baicen Xiao, Mayoore Jaiswal, and Radha Poovendran. 2017. On the limitation of convolutional neural networks in recognizing negative images. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 352–358.

[14] Forrest Huang, John F Canny, and Jeffrey Nichols. 2019. Swire: Sketch-based user interface retrieval. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10.

[15] Laurent Itti and Christof Koch. 2000. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research* 40, 10-12 (2000), 1489–1506.

[16] Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* 20, 11 (1998), 1254–1259.

[17] Albert Jimenez, Jose M Alvarez, and Xavier Giro-i Nieto. 2017. Class-weighted convolutional features for visual instance search. *arXiv preprint arXiv:1707.02581* (2017).

[18] Jussi PP Jokinen, Zhenxin Wang, Sayan Sarcar, Antti Oulasvirta, and Xiangshi Ren. 2020. Adaptive feature guidance: Modelling visual search with graphical layouts. *International Journal of Human-Computer Studies* 136 (2020), 102376.

[19] David E Kieras and Anthony J Hornof. 2014. Towards accurate and practical predictive models of active-vision-based visual search. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 3875–3884.

[20] Zakaria Laskar and Juho Kannala. 2017. Context aware query image representation for particular object retrieval. In *Scandinavian Conference on Image Analysis*. Springer, 88–99.

[21] Yuan Liang, Hsuan Wei Fan, Zhujun Fang, Leiying Miao, Wen Li, Xuan Zhang, Weibin Sun, Kun Wang, Lei He, and Xiang 'Anthony' Chen. 2020. OralCam: Enabling Self-Examination and Awareness of Oral Health Using a Smartphone Camera. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. DOI: http://dx.doi.org/10.1145/3313831.3376238

[22] David Ebri Mars, Hanwei Wu, Haopeng Li, and Markus Flierl. 2015. Geometry-based ranking for mobile 3D visual search using hierarchically structured multi-view features. In *2015 IEEE International Conference on Image Processing (ICIP)*. 3077–3081. DOI: http://dx.doi.org/10.1109/ICIP.2015.7351369

[23] Fabrice Matulic, Riku Arakawa, Brian Vogel, and Daniel Vogel. 2020. PenSight: Enhanced Interaction with a Pen-Top Camera. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. DOI: http://dx.doi.org/10.1145/3313831.3376147

[24] Brian McElree and Marisa Carrasco. 1999. The temporal dynamics of visual search: evidence for parallel processing in feature and conjunction searches. *Journal of Experimental Psychology: Human Perception and Performance* 25, 6 (1999), 1517.

[25] Microsoft. 2018. Sketch2Code. (2018). https://sketch2code.azurewebsites.net/

[26] Eva Mohedano, Kevin McGuinness, Xavier Giró-i Nieto, and Noel E O'Connor. 2018. Saliency weighted convolutional features for instance search. In *2018 international conference on content-based multimedia indexing (CBMI)*. IEEE, 1–6.

[27] Enkhbold Nyamsuren and Niels A Taatgen. 2013. Pre-attentive and attentive vision module. *Cognitive systems research* 24 (2013), 62–71.

[28] Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017).

[29] Johannes Pfau, Jan David Smeddinck, Ioannis Bikas, and Rainer Malaka. 2020. Bot or Not? User Perceptions of Player Substitution with Deep Player Behavior Models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. DOI: http://dx.doi.org/10.1145/3313831.3376223

[30] Alex Robinson. 2019. Sketch2code: Generating a website from a paper mockup. (2019).

[31] Jiye Shen, Eyal M Reingold, and Marc Pomplun. 2003. Guidance of eye movements during conjunctive visual search: the distractor-ratio effect. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale* 57, 2 (2003), 76.

[32] Luke Taylor and Geoff Nitschke. 2017. Improving deep learning using generic data augmentation. *arXiv preprint arXiv:1708.06020* (2017).

[33] Leong-Hwee Teo, Bonnie John, and Marilyn Blackmon. 2012. CogTool-Explorer: A model of goal-directed user exploration that considers information layout. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2479–2488.

[34] A. Treisman and G. Gelade. 1980. A feature-integration theory of attention. *Cognitive Psychology* 12 (1980), 97–136.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[36] Francesco Visin, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron Courville, and Yoshua Bengio. 2015. Renet: A recurrent neural network based alternative to convolutional networks. *arXiv preprint arXiv:1505.00393* (2015).

[37] Tonghe Wang, Yang Lei, Yabo Fu, Jacob F Wynne, Walter J Curran, Tian Liu, and Xiaofeng Yang. 2021. A review on medical imaging synthesis using deep learning and its clinical applications. *Journal of Applied Clinical Medical Physics* 22, 1 (2021), 11–36.

[38] Jeremy M Wolfe and Todd S Horowitz. 2017. Five factors that guide attention in visual search. *Nature Human Behaviour* 1, 3 (2017), 1–8.

[39] Xiaoli Wu, Tom Gedeon, and Linlin Wang. 2018. The analysis method of visual information searching in the human-computer interactive process of intelligent control system. In *Congress of the International Ergonomics Association*. Springer, 73–84.

[40] Arianna Yuan and Yang Li. 2020. Modeling Human Visual Search Performance on Realistic Webpages Using Analytical and Deep Learning Methods. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[41] Zhilu Zhang and Mert R Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *arXiv preprint arXiv:1805.07836* (2018).

[42] Li Zhaoping and Uta Frith. 2011. A clash of bottom-up and top-down processes in visual search: The reversed letter effect revisited. *Journal of Experimental Psychology: Human Perception and Performance* 37, 4 (2011), 997.